SRI's 1998 Broadcast News System – Toward Faster, Better, Smaller Speech Recognition[*]

Ananth Sankar, Ramana Rao Gadde and Fuliang Weng

Speech Technology and Research Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

ABSTRACT

We describe several new research directions we investigated toward the development of our broadcast news transcription system for the 1998 DARPA H4 evaluations. Our goal was to develop significantly faster and smaller speech recognition systems without degrading the word error rate of our 1997 system. We did this through significant algorithmic research creating various new techniques. A sample of these techniques was used to put together our 1998 broadcast news system, which is conceptually much simpler, faster, and smaller, but gives the same word error rate as our 1997 system. In particular, our 1998 system is based on a simple phonetically tied mixture (PTM) model with a total of only 13,000 Gaussians, as compared to a 67,000-Gaussian state-clustered system we used in 1997.

1. Introduction

One of our main goals in 1998 was to significantly increase speed and decrease model size, while maintaining or improving accuracy. These goals are difficult to achieve simultaneously because of inherent trade-offs. Decreasing the number of system parameters will typically degrade accuracy. Similarly increasing the speed by eliminating search passes, or decreasing the pruning beamwidth during the decoding stage, will degrade accuracy. We decided, therefore, that to achieve simultaneous improvements in speed, size, and accuracy, we would have to significantly alter our approach by focusing on novel algorithms. We developed and studied several new algorithms for acoustic modeling, adaptation, and lattice generation. A sample of these methods was incorporated into our 1998 broadcast news system. However, we were unable to incorporate all the methods we developed because of time and resource constraints. The resulting system was significantly simpler and faster than our 1997 system.

In this paper, we summarize the various new techniques we developed, along with experimental results. We then briefly describe our 1998 broadcast news evaluation system and give experimental results for 1996 H4 partitioned evaluation (PE) development test data and the 1998 DARPA H4 evaluation data. Finally, we summarize work we did after the 1998 evaluation to further improve our system performance.

2. Improved Parameter Tying

Most current state-of-the-art speech recognition systems are based on state-clustered hidden Markov models (HMMs). However, the significant overlap of state clusters in acoustic space leads to potential problems. The data in the cluster overlap regions is divided between clusters, giving less robust Gaussian estimates. Gaussians from each state cluster may also overlap with each other, causing redundancy and a waste of parameters. These modeling problems can be easily handled by decreasing the number of clusters and appropriately increasing the number of Gaussians per cluster [1]. We also expect a recognition speed-up because of significant savings in Gaussian computation due to the smaller variances of the Gaussians [1] when state clusters are merged. In our approach, we used a PTM system with only 40 state clusters, and a large number of Gaussians per class.

Table 1 shows that the new PTM approach gave significantly lower word error rate (WER) than a state-clustered system on two Wall Street Journal (WSJ) test sets, and a North American Business News (NABN) test set, using a 20,000-word bigram language model (LM). The state-clustered system had 937 clusters, while the PTM system used 40 phone classes. Both systems had a total of about 30,000 Gaussians.

System	Word Error Rate (%)		
	WSJ1	WSJ2	NABN
State-clustered	21.65	14.08	18.29
PTM	20.49	12.58	16.78

Table 1: Word error rates for different levels of tying

Figures 1 and 2 plot the word error rate against the number of Gaussians computed and the recognition time, respectively. Each point on the curve is for a different value of the pruning beamwidth in our Viterbi search. At a word error rate of 22%, the PTM system computes half the number of Gaussians. Also, the PTM system achieves this accuracy with a smaller pruning beamwidth, thus resulting in significantly fewer active hypotheses in the search. The resulting speed-up for the PTM

[*] This work was sponsored by DARPA through the Naval Command and Control Ocean Surveillance Center under contract N66001-94-C-6048.

system is a factor of 5.

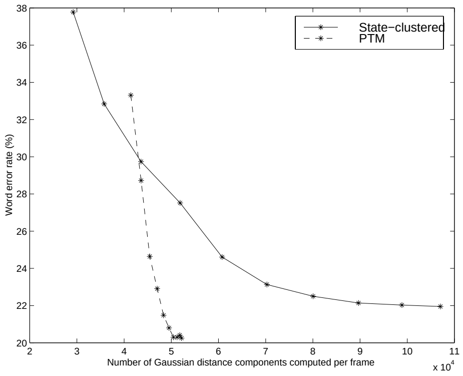

Figure 1: Word error vs. number of Gaussian distance components computed

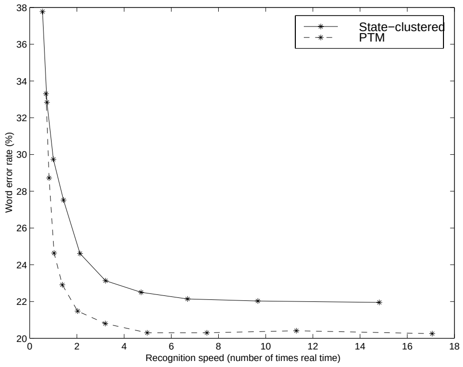

Figure 2: Word error vs. recognition speed

3. Phone-level Gaussian Clustering

In the PTM approach, we must use a much larger number of Gaussians per state cluster than in previous state-clustered systems. However, some phone classes may have very little acoustic variability and thus may need only a few Gaussians for good modeling. For example, the nasal $/ng/$ is less variable than the unvoiced stop $/t/$.

We exploited this fact by developing a per-phone Gaussian clustering algorithm that automatically determines the number of Gaussians per phone based on the measured acoustic variability. To measure a phone's acoustic variability, we agglomeratively cluster the HMM states for each phone, using a weighted-by-counts entropy distance between the mixture weight distributions of each state [2]. Clustering is stopped when the average distance reaches a pre-specified relative threshold. The number of resulting state clusters is a measure of a phone's acoustic variability. In our study, the number of Gaussians for a phone was proportional to the acoustic variability, with a pre-set minimum and maximum number of Gaussians.

Table 2 shows the word error rate on the 1996 H4 PE development test set and the number of Gaussians for three different female models trained on the first 100 hours of H4 training data. These recognition runs used a 48,000-word bigram LM. 1997−eval is a state-clustered model we used for the 1997 DARPA H4 evaluations. PTM−1788 is a PTM model with 1788 Gaussians per phone class, and Clustered−PTM is a model created by applying the per-phone Gaussian clustering algorithm to PTM−1788. From the table, we see that the PTM and state-clustered systems gave the same word error rate. A factor of 5 reduction in the number of Gaussians was achieved using the per-phone Gaussian clustered PTM model, with no difference in the word error rate. The drastic reduction in Gaussians also decreases the amount of computation during recognition.

Model	Word (%) Error	Number of Gaussians
1997-eval	39.4	67,200
PTM-1788	39.7	69,732
Clustered-PTM	39.3	12,758

Table 2: Word error rates and number of Gaussians for different models

4. Mixture Weight Reduction

One problem with our PTM modeling approach is that the mixture weight distributions for each state can become very large because of the large number of Gaussians per phone class. However, since only a few Gaussians will be active for each HMM state, we can more efficiently represent the weights. We examined two recently published schemes to reduce the number of mixture weights in our PTM models [3]. In the first, called the "Zeroing" scheme, we set all mixture weights below a threshold to zero and renormalize the mixture weights. In the second, called the "Averaging" scheme, we set each mixture weight below the threshold to a value equal to the average of all mixture weights below the threshold.

Experimental results showed that the Zeroing scheme worked for small thresholds, but rapidly deteriorated as the threshold was increased. However, the Averaging scheme maintained a low word error rate even for large thresholds, resulting in a factor of 16 reduction in the number of mixture weights with no degradation in accuracy.

5. Tied-transform HMMs

We developed a new modeling and training algorithm called the tied-transform (T^2) HMM, which gives robust estimates for systems with a large number of Gaussians [4]. The basic idea is illustrated in Figure 3, which shows an HMM state-cluster tree. Suppose our goal is to train an HMM for the larger number of state clusters N. However, we do not have enough data to robustly estimate each Gaussian in this large system. We solve this problem by training an HMM for the smaller number of state clusters M, for which we assume that we have enough data to robustly estimate each Gaussian. The Gaussians in the state clusters of the larger HMM are transformed versions of the Gaussians in the ancestor state clusters in the smaller HMM, where the transformations are estimated as in maximum-likelihood adaptation [5, 6, 7, 8].

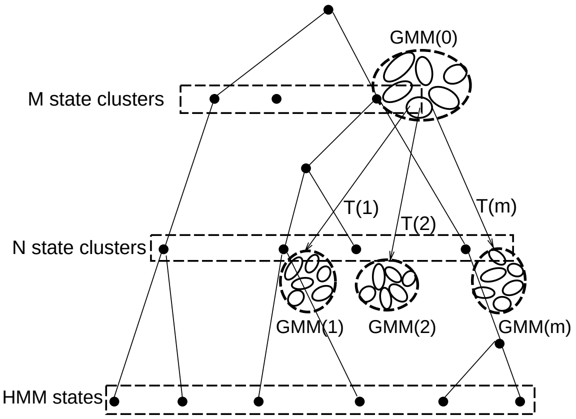

Figure 3: Illustration of T^2-HMM

Experimental results showed that the T^2-HMM method gave significant improvement in word accuracy in our experiments with state-clustered HMMs [4]. However, we did not use this algorithm in our 1998 broadcast news evaluation system because of a lack of time.

6. Fast Adaptation

To speed up our system, we developed various engineering solutions to decrease the computation cost for our maximum-likelihood (ML) transformation-based adaptation algorithms. We also proposed a new adaptation algorithm called “Basis Transform Adaptation”, which can be advantageously used when the amount of adaptation data is small.

Most ML transformation-based adaptation algorithms [5, 6, 7, 8] involve three steps: (1) computing sufficient statistics for all Gaussians in the model, (2) estimation of transform statistics from the Gaussian statistics, and (3) estimation of the transform from the transform statistics. We examined the computations involved in these three steps and observed that the first two steps take most of the adaptation time. Hence, we investigated ways to reduce the computation for each of these steps.

Use of Viterbi alignments instead of the usual forward-backward algorithm to compute the Gaussian statistics gave a factor of 4 speed-up with no loss of accuracy. To reduce the time for estimation of transform statistics, we used a threshold on the Gaussian counts to decide which Gaussians would be used. Only Gaussians with counts higher than the threshold are used to compute transform statistics. We experimented with a number of thresholds and found that a threshold of 0.1 on the Gaussian counts reduced the time by 40% with no loss of accuracy. Higher thresholds reduced the time further, but degraded accuracy. These results are shown in Table 3.

Threshold	WER (%)	Adaptation Speed(X RT)
0.0	29.18	1.6
0.1	29.16	1.0
0.5	29.30	0.9
1.0	29.58	0.75

Table 3: Results of Gaussian thresholding

A different scheme we investigated used smaller models to compute the adaptation transforms and applied them on larger models. Since the smaller models have fewer Gaussians, adaptation time is decreased. In our experiments, we were able to reduce the adaptation time by nearly 45% with practically no change in accuracy with some model combinations. However, as these results were not consistent across all model combinations, we did not use this scheme in our 1998 evaluation system.

We also developed a new method called “Basis transform adaptation”, which has significant speed advantages over ML transformation-based techniques like ML linear regression (MLLR). The adaptation transform for the test speaker is a weighted combination of a set of basis transforms. The basis transforms are trained using the training data; during testing, we only estimate the small number of combination weights. Thus, this approach can give much faster adaptation. In our approach, we estimated the combination weights by maximizing the likelihood of the adaptation data.

We compared the performance of basis transform adaptation with MLLR adaptation using a state-clustered HMM model with 252 state clusters and 128 Gaussians per state cluster. The models were trained on a 71-speaker subset of the WSJ male training set. We estimated affine transforms of the Gaussian means for 46 training speakers and used them as basis transforms. We did supervised adaptation on 10 test speakers using the 40 common adaptation sentences. The results, in Table 4, show that the basis transform method performs as well as MLLR for small amounts of adaptation data; however, as more data becomes available, MLLR performs better.

We then experimented with unsupervised transcription-based adaptation done on each sentence of the same test set. The results, in Table 5, again show that the basis-transform approach and MLLR give the same results. The advantage of the basis-transform adaptation algorithm over MLLR is the

Model	#adapt. sentences(#transforms)				
	1(1)	2(1)	5(2)	10(5)	20(50)
SI	22.8	22.8	22.8	22.8	22.8
MLLR	22.4	21.7	21.0	20.9	20.2
Basis Transform	21.8	21.9	21.8	21.2	21.3

Table 4: Comparison of MLLR and Basis Transform adaptation (supervised adaptation)

speed-up we expect because only a very few parameters are estimated during testing.

Model	WER(%)
SI	22.8
MLLR	21.7
Basis Transform	21.8

Table 5: Comparison of MLLR and Basis Transform adaptation (unsupervised adaptation)

7. Lattice Algorithms

In SRI's 1998 multipass broadcast news transcription system, word lattices are used as an intermediate representation. To achieve efficiency and accuracy, we want to have small lattices with a low lattice error rate. We tested two bigram lattice reduction algorithms that we recently developed. These are the exact and approximate reduction algorithms [9], which gave a 50% and 67% size reduction, respectively, over the original bigram lattices. The approximate reduction algorithm also gave a 6% and 34% lower lattice error for F0 and F1 conditions. Compared with the standard finite state machine (FSM) determinization and minimization algorithms implemented by AT&T, our two algorithms produced lattices with 8% and 39% smaller sizes. These results are shown in Table 6. For the 1998 evaluations, we used the exact reduction algorithm.

	SIZE		LER	
	F0	F1	F0	F1
Baseline	11624	16208	3.3%	10.0%
FSM Det/Min	6417	8715	3.3%	10.0%
Exact Red	6129	7797	3.3%	10.0%
Approx Red	4318	4985	3.1%	6.6%

Table 6: Sizes and lattice error rates of the reduction algorithms

Trigram LMs were incorporated by expanding the reduced bigram lattices to trigram lattices, using our recently developed compact trigram expansion algorithm [10]. Comparative experimental results show that our compact trigram expansion algorithm gives more than 50% smaller lattices than those generated by the AT&T FSM tools [9]. In addition, our compact trigram expansion is 10 times faster than the conventional trigram expansion with no accuracy degradation.

8. Confidence-based Language Modeling

Error analysis on Switchboard data shows that the LM is more likely to predict a word incorrectly when the previous word is incorrect than if the previous word were correct [11]. This intuitive fact can be exploited to develop better LMs. If we knew a hypothesized word was incorrect, we would not compute the probability for the next word conditioned on it. Instead, it would make more sense to back off to the unigram probability if we knew the previous word was wrong. Automatically detecting which words are incorrect is challenging. One approach is to use acoustic confidence scores as evidence to judge the correctness of a word. The proposed confidence-based language model (CBLM) computes the word n-gram probability as follows. For simplicity, we use a trigram LM as an example:

$$\begin{aligned} P^*(w_3 \mid w_2 w_1) &= P(X_2 = 1, X_1 = 1) * P(w_3 \mid w_2 w_1) \\ &+ P(X_2 = 1, X_1 = 0) * P(w_3 \mid w_2) \\ &+ P(X_2 = 0) * P(w_3) \end{aligned}$$

where, $X_i,\ i = 1, 2$, are random variables indicating the correctness of w_i. Thus,

$$X_i = \begin{cases} 1 & \text{if } w_i \text{ is correct} \\ 0 & \text{if } w_i \text{ is incorrect.} \end{cases}$$

We can interpret the probabilities as confidence measures for the correctness of the hypothesized words. To get an estimate of the maximum improvement from this approach, we designed a “cheating” experiment [12]. In this experiment, we assume that the correctness of all the words in the n-best hypotheses is known, therefore giving perfect confidence scores. Based on this information, the trigram probability is used if both previous words are correct; the bigram probability is used if the nearest word in the history is correct; and otherwise, the unigram probability is used. A similar experiment with bigrams was also conducted to observe the consistency of the results. These LMs were used to attach LM scores to the entries in the n-best list, and the hypothesis with the maximum combined acoustic and LM score was selected. The bigram and trigram results are shown in Table 7. In both cases, more than 1% absolute improvement was obtained on the 1996 H4 development test data. The n-best error rate was about 18%. It is possible that we would get even higher gains if we used lattices to do this experiment.

Model	conventional	confidence-based
bigram	36.8%	35.7%
trigram	33.4%	32.3%

Table 7: Word error rates of confidence-based n-gram

This initial experiment shows a potentially moderate improvement, using CBLM. Future research is needed to apply real acoustic confidence measures on lattices to further verify the idea.

9. Confidence-based Optimization

In our 1997 H4 experiments, we observed that sentences with smaller lattices usually have lower word error rates, probably because smaller lattices indicate lower confusability. We conducted a set of experiments on the 1996 H4 PE development data to see whether tuning LM weights based on lattice sizes would give us any gain.

Two sets of experiments were designed, each dividing the whole development test set into four subgroups. The first experiment divides the test set based on normalized lattice sizes. The normalized lattice size was estimated by dividing the number of transitions by the number of nodes in a lattice. The other experiment randomly divides the data into four subgroups. A 0.3% absolute improvement was derived using the lattice size to determine separate LM weights. However, tuning the LM weights to four randomly selected lattice groups gave an improvement of 0.18%. The LMs tuned to the size-based partitions did not perform much better than those tuned to random partitions. This could imply that using multiple partitions has the effect of tuning the LM to the development data, and that these results may not carry through to evaluation data. We did not pursue this work further at this time.

10. Broadcast News System and Experimental Results

We will not give a detailed description of the system, but refer the reader to the description available in NIST's Web page [13]. Instead, we list the novel features and algorithms used in our 1998 system along with the sections in this paper which describe them:

1. Novel parameter tying (Section 2)
2. Per-phone Gaussian clustering (Section 3)
3. Adaptation speed-ups (Section 6)
4. New lattice generation (Section 7)

We used the new acoustic modeling techniques to configure a per-phone Gaussian clustered PTM system with a total of only 13,000 Gaussians. We developed two systems–a hub system for which processing time was not a constraint, and a spoke system, which ran in 10 times real time. The main difference between the hub and spoke systems was tighter pruning and the elimination of one acoustic adaptation and recognition stage for the spoke.

Table 8 gives the word error rates on the 1996 H4 PE development test set using our 1997 evaluation system, the 1998 hub system, and the 1998 10 times real-time spoke system. We see that the 1997 and 1998 systems gave almost identical error rates. However, our 1998 13,000-Gaussian PTM system is clearly far simpler than our 67,000-Gaussian state-clustered system of 1997. Table 9 gives the word error rates for the two 1998 H4 evaluation data test sets (S1 and S2) using our hub and spoke systems. The 10 times real-time system gave relatively minor degradation compared to the hub system.

System	Word Error (%)
1997 SRI eval	26.1
1998 SRI H4 Hub	26.7
1998 SRI H4 10XRT Spoke	28.8

Table 8: Word error on 1996 PE development test set

System	Word Error (%)	
	S1	S2
1998 SRI H4 Hub	22.1	20.1
1998 SRI H4 10XRT Spoke	23.4	22.2
Degradation over Hub system	5.9%	10.4%

Table 9: Word error on 1998 H4 evaluation test set

11. Post-evaluation Experiments

Based on our experimental results, we believe we have made very good progress toward our goal of developing significantly faster and smaller systems with no degradation in word error. However, because of a lack of time, we were unable to try various existing techniques published in the literature, and to tune our system to attain its lowest possible error rate. Also, we used only the first 100 hours of training data to train our acoustic models, rather than the available 200 hours of data. After the evaluation, we addressed some of these issues. In particular, we

1. Trained our system on 200 hours of data
2. Tuned the number of parameters more carefully

3. Implemented a diagonalizing tied covariance transform [14]

4. Used BBN's 1998 evaluation segments [15] to evaluate our own segmentation algorithm

We evaluated our segmentation algorithm by running our spoke system on the 1998 evaluation data using our own segments and those we got from BBN. We found that using BBN's segments gave us 1.4% and 0.9% absolute improvements in word error rate for the S1 and S2 evaluation test sets, respectively, showing that there is room for improvement in our segmentation algorithm.

We then configured a 30,000-Gaussian PTM system using the same approach used for the evaluation system, but with the additional post-evaluation improvements. We used this system to run a 10 times real time spoke test on the evaluation data. Table 10 gives the word error rates for the 10 times real time spoke task for our evaluation system, and the post-evaluation system. A significant improvement was achieved using the post-evaluation system. We note that this system has more Gaussians than the evaluation system, but the number of Gaussians is still significantly smaller than other state-of-the-art systems.

System	Word Error (%)	
	S1	S2
1998 SRI H4 10XRT Spoke	23.4	22.2
Post-eval 10XRT Spoke	21.3	19.7
Improvement over eval system	9.0%	11.3%

Table 10: Comparison of evaluation and post-evaluation Spoke systems

12. Summary and Conclusion

We developed many new techniques for the 1998 DARPA H4 evaluations. Our main focus was to drastically decrease recognition time and model size while not compromising the accuracy. Toward this goal, we made good progress, creating a simple 13,000-Gaussian PTM system that performed as well as our more complex 1997 state-clustered system with 67,000 Gaussians. By using BBN's segments, and a larger 30,000-Gaussian PTM system trained on all the available training data, a further improvement of about 10% was achieved. The new technologies that were used include a new parameter tying method, a per-phone Gaussian clustering algorithm, fast adaptation algorithms, and new lattice reduction and representation algorithms.

Acknowledgements

We thank Andreas Stolcke for training the language models we used. We also thank Long Nguyen of BBN and Ramesh Gopinath of IBM for providing their 1998 evaluation segments for our post-evaluation analysis.

References

1. Ananth Sankar, "A New Look at HMM Parameter Tying for Large Vocabulary Speech Recognition," in *Proceedings of ICSLP*, (Sydney, Australia), 1998.
2. V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 4, pp. 281–289, 1996.
3. S. Gupta, F. Soong, and R. Hami-Cohen, "Quantizing Mixture Weights in a Tied-Mixture HMM," in *Proceedings of ICSLP*, pp. 1828–1831, 1996.
4. Ananth Sankar, "Robust HMM Estimation with Gaussian Merging-Splitting and Tied-Transform HMMs," in *Proceedings of ICSLP*, (Sydney, Australia), 1998.
5. A. Sankar and C.-H. Lee, "Stochastic Matching for Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 1, pp. 124–125, August 1994.
6. A. Sankar and C.-H. Lee, "A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 190–202, May 1996.
7. V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker Adaptation Using Constrained Reestimation of Gaussian Mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.
8. C. J. Legetter and P. C. Woodland, "Flexible Speaker Adaptation Using Maximum Likelihood Linear Regression," in *Proceedings of the Spoken Language Systems Technology Workshop*, pp. 110–115, 1995.
9. F. Weng, A. Stolcke, and A. Sankar, "Efficient Lattice Representation and Generation," in *Proceedings of ICSLP*, (Sydney, Australia), 1998.
10. F. Weng, A. Stolcke, and A. Sankar, "New Developments in Lattice-based Search Strategies in SRI's H4 system," in *Proceedings of DARPA Speech Recognition Workshop*, (Lansdowne, VA), February 1998.
11. C. Neti, S. Roukos, and E. Eide, "Word-Based Confidence Measures as a Guide for Stack Search in Speech Recognition," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997.
12. Hierarchical Consistency Modeling for Next-Generation Speech Recognition–quarterly progress report covering the period March 15, through June 1, 1997, *submitted to DARPA*.
13. ftp://jaguar.ncsl.nist.gov/csr98/
h4e_98_official_scores_981125/readme.html.
14. Mark Gales, "Semi-Tied Full-Covariance Matrices for Hidden Markov Models," Tech. Rep. CUED/F-INFENG/TR 287, Cambridge University, April 1997.
15. S. Matsoukas, L. Nguyen, J. Davenport, J. Billa, F. Richardson, D. Liu, R. Schwartz, and J. Makhoul, "The 1998 BBN BYBLOS Primary System Applied to English and Spanish Broadcast News Transcription," in *Proceedings of the DARPA Broadcast News Workshop*, (Washington, D.C.), 1999.